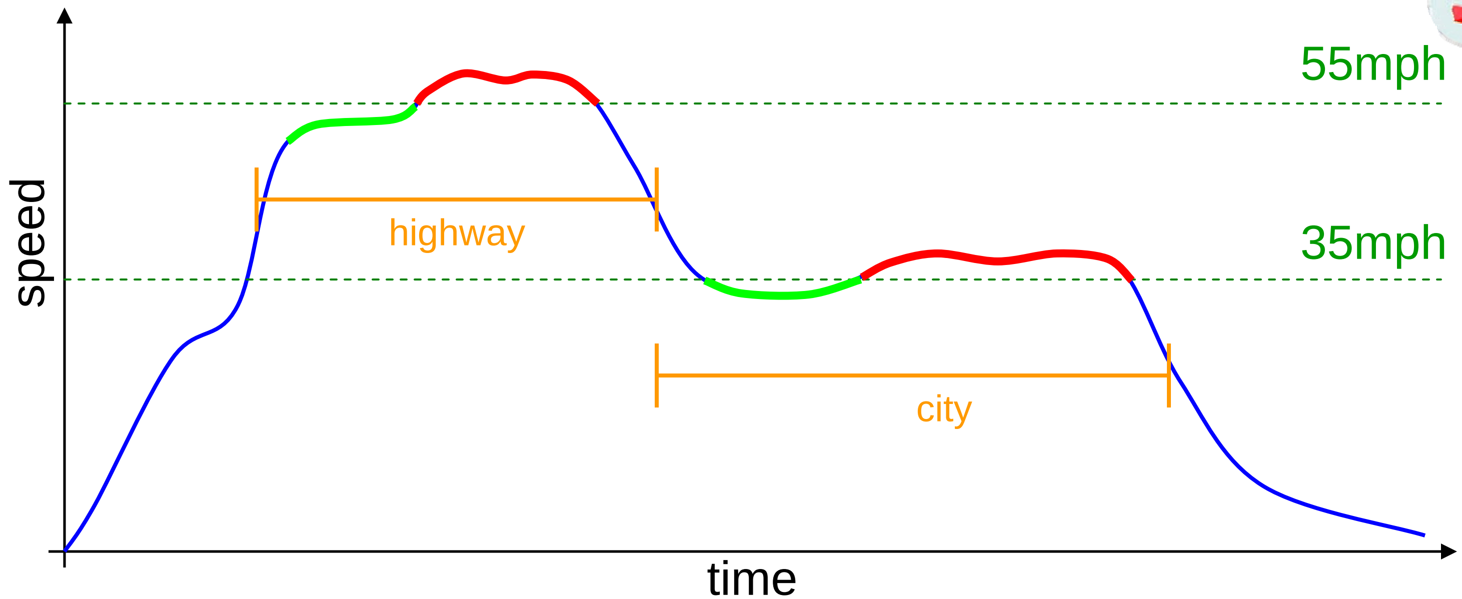
Time Series Compressibility and Privacy

Spiros Papadimitriou*
Feifei Li+
George Kollios+
Philip S. Yu*

*IBM TJ Watson
+Boston University

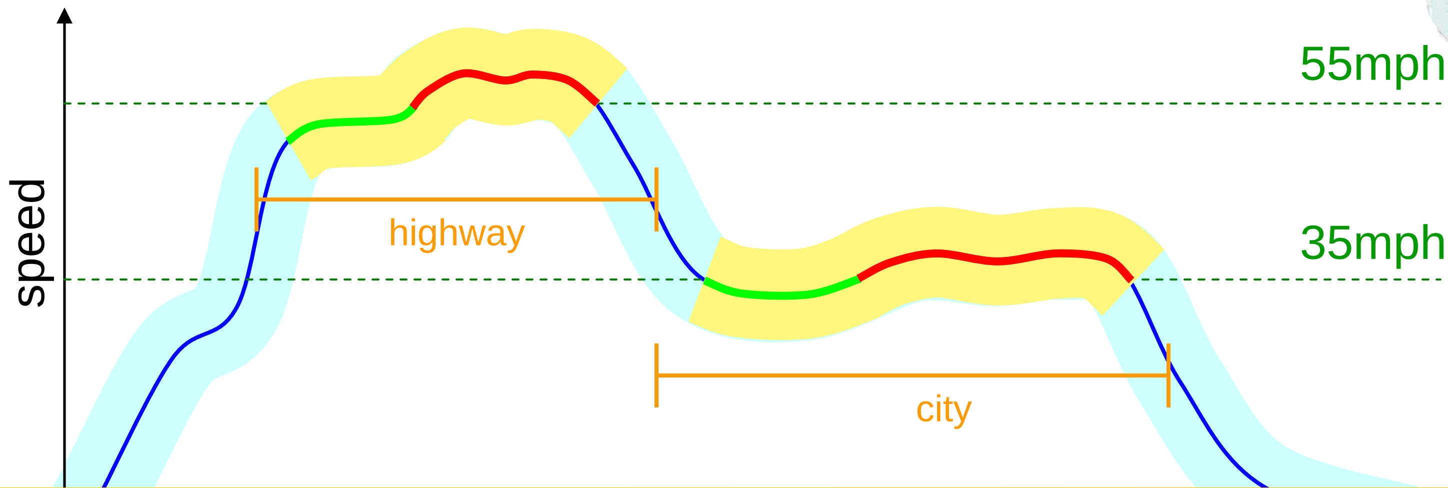
Intuition / Motivation

- Introduce uncertainty about individual values, while still allowing interesting pattern mining



Intuition / Motivation

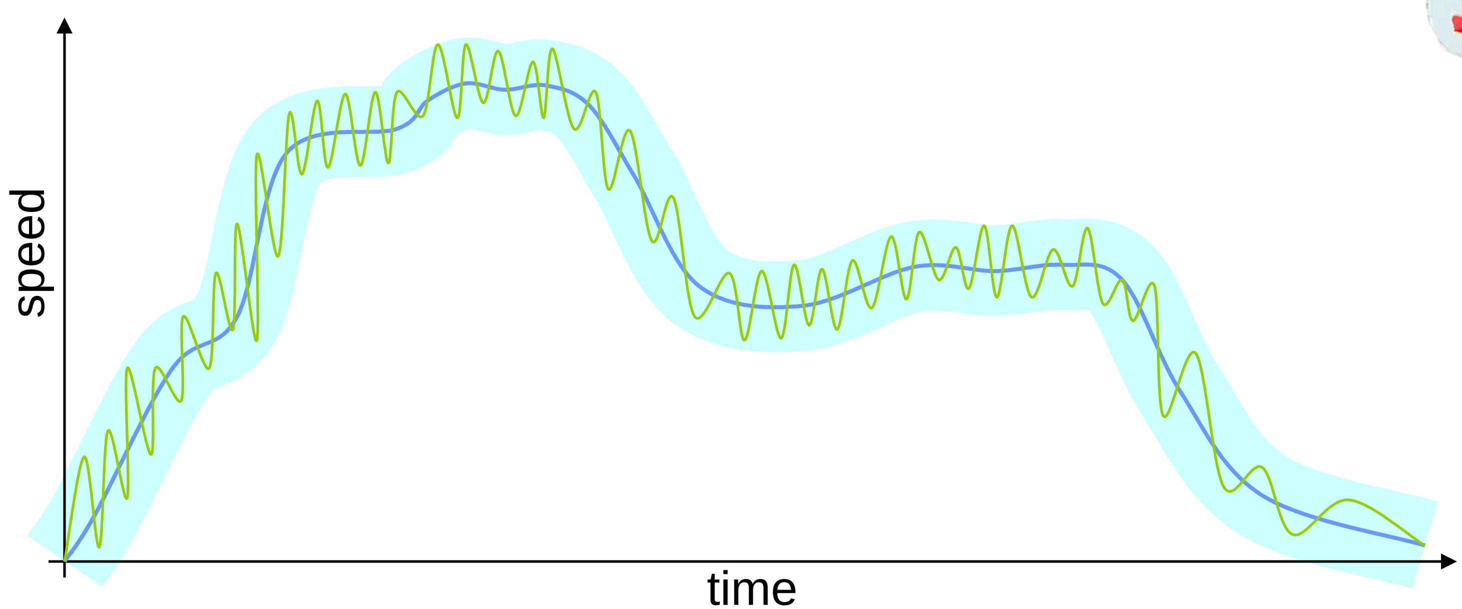
- Introduce uncertainty about individual values, while still allowing interesting pattern mining



Need to publish *some* value within the band: which one?

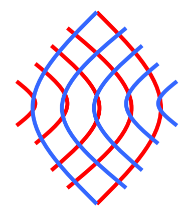
Random (white noise) ?

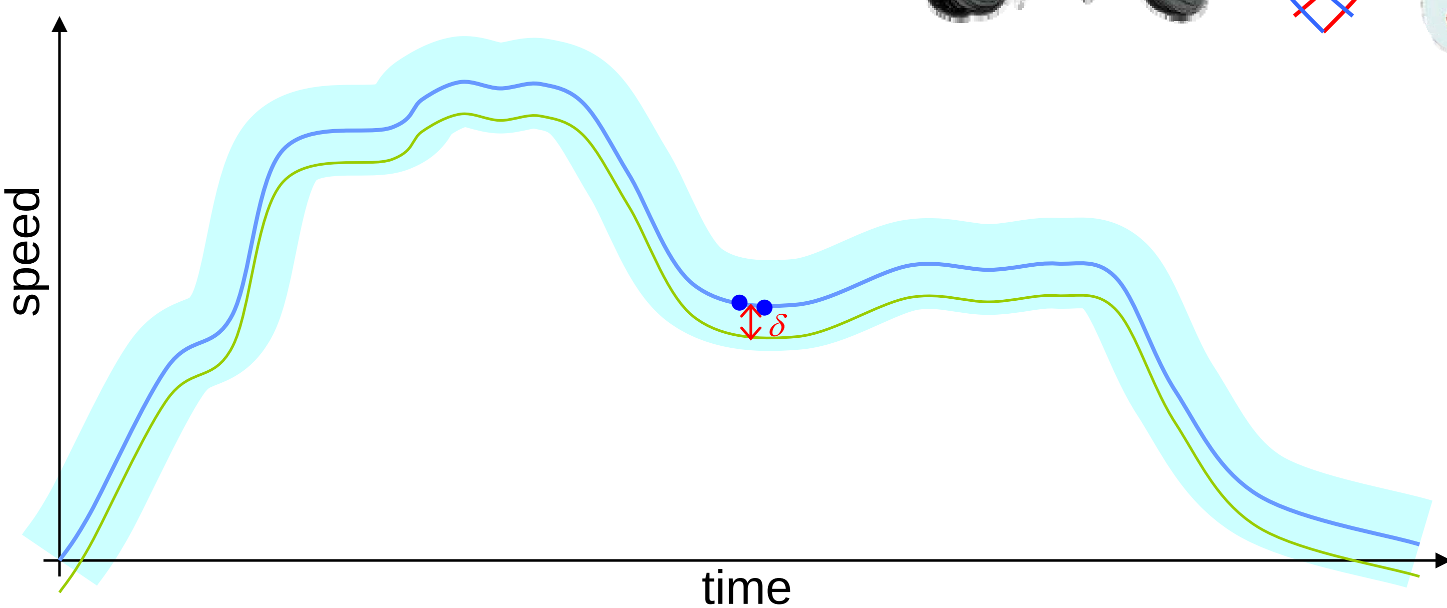
- Completely random permutation?
- Cars (typically) don't drive like this
  $\Rightarrow$ Noise can be filtered out



Deterministic ?

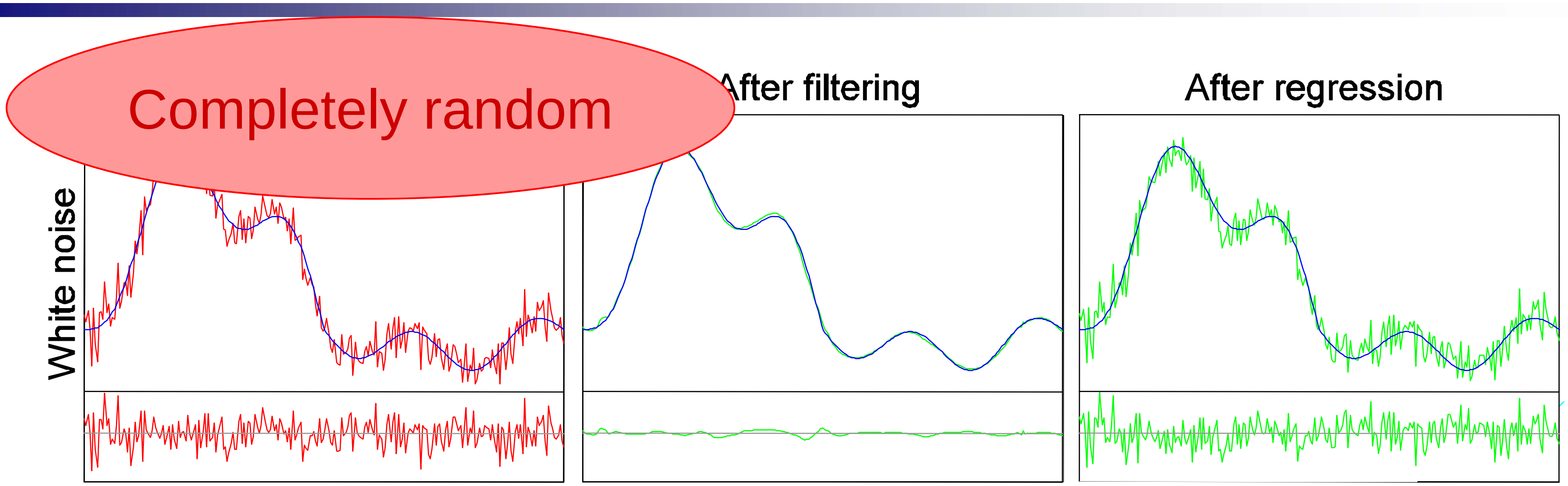
- Completely “deterministic” permutation?
- True value leaks



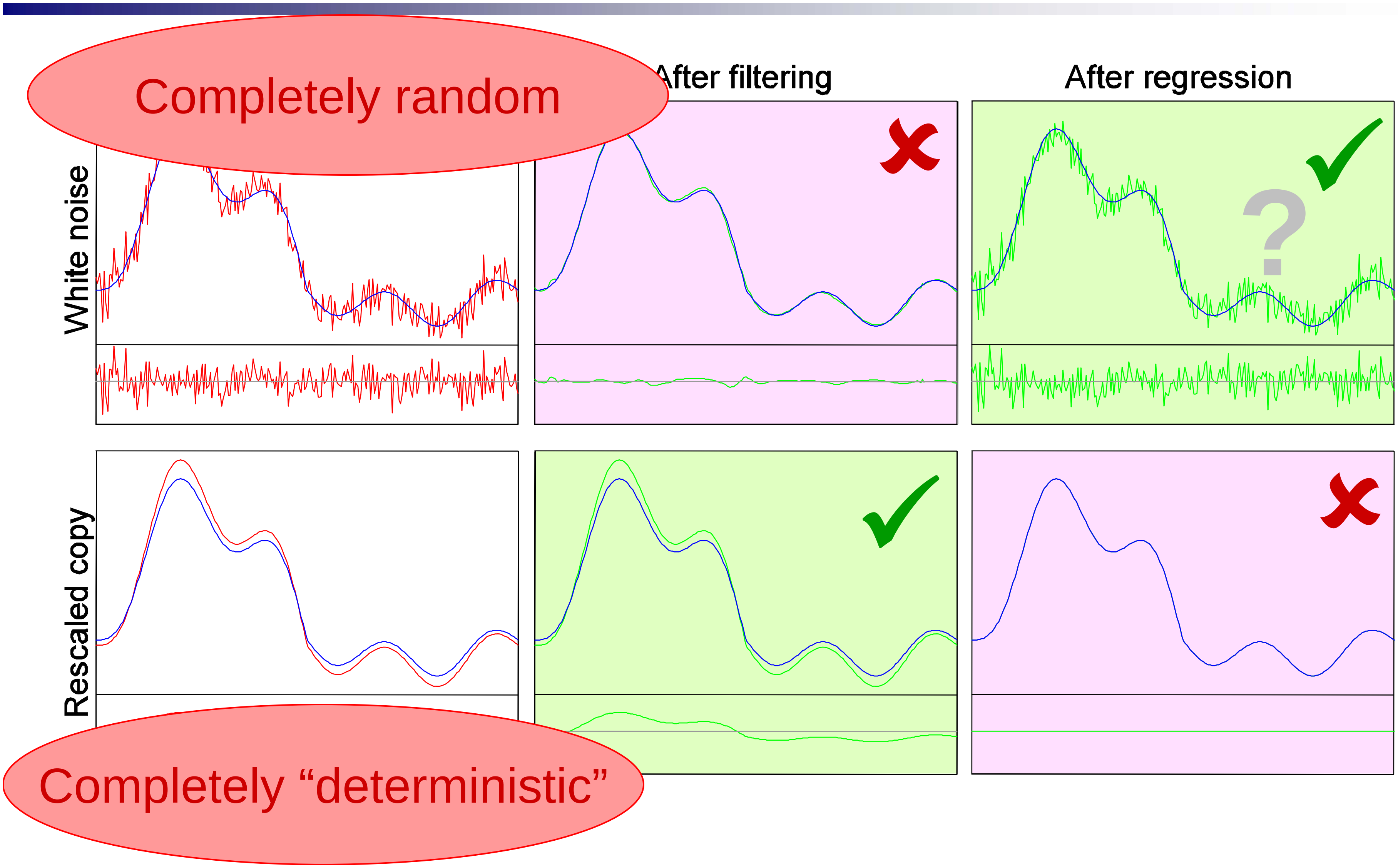


First extreme case

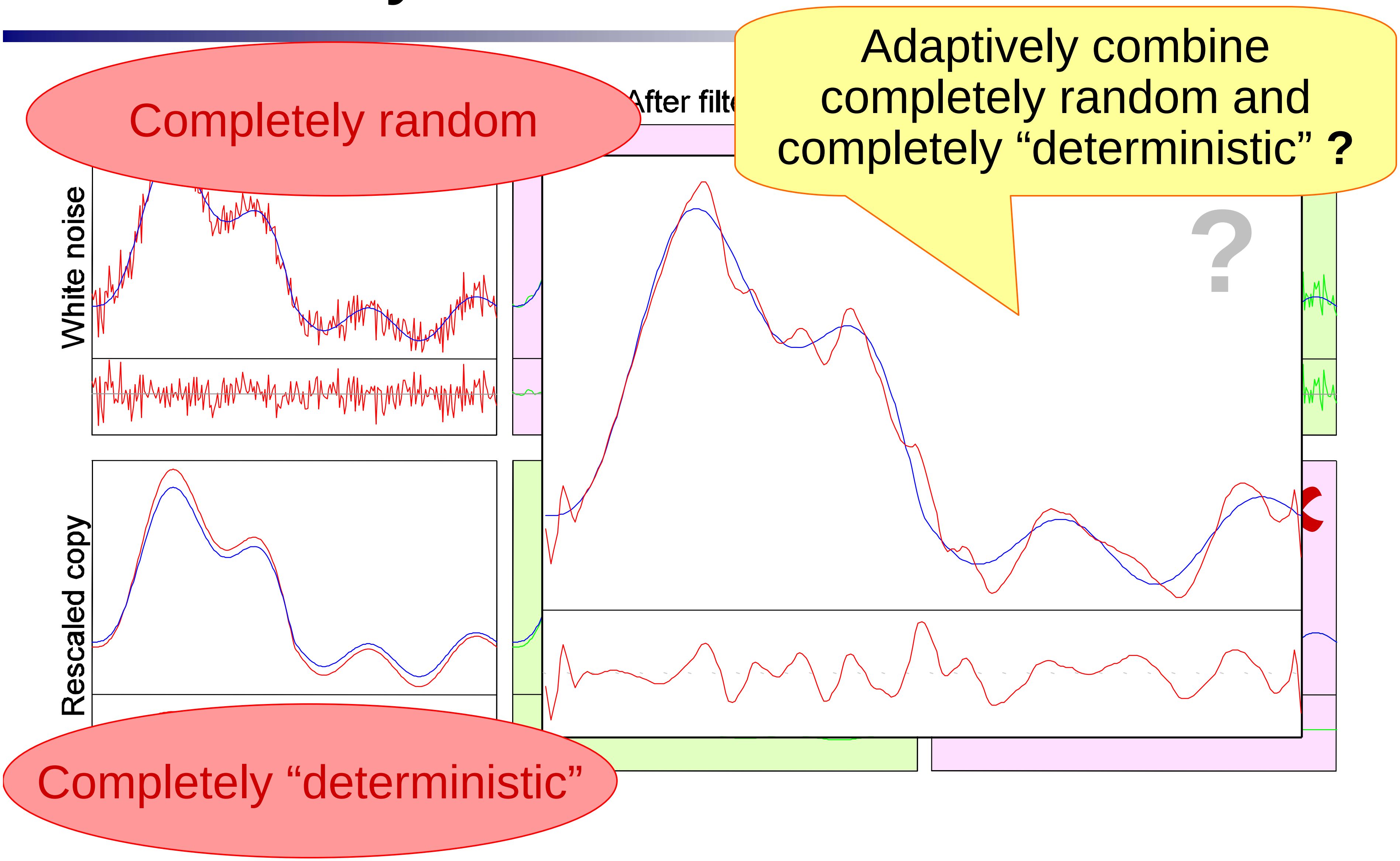
White noise



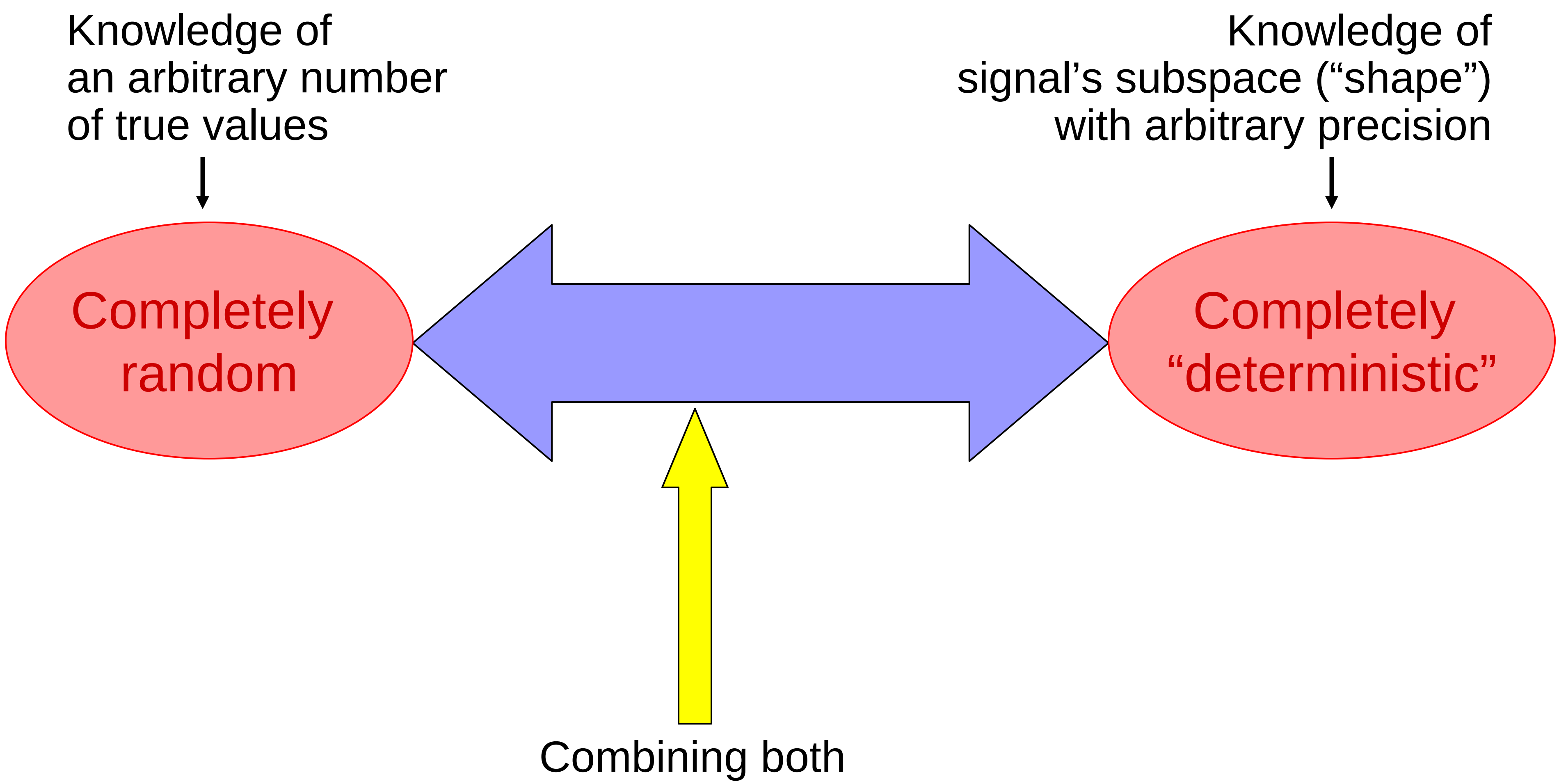
Summary of extreme cases



Summary of extreme cases



Main challenge



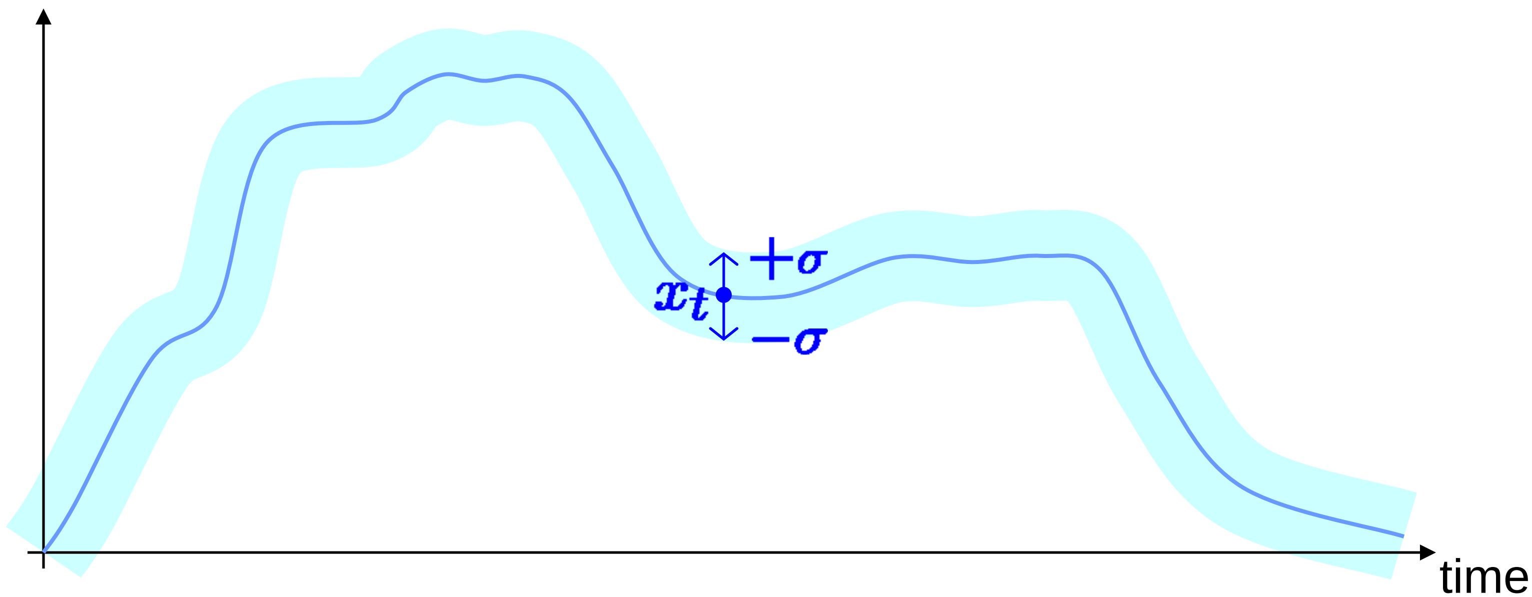
Goals

- Partial “information hiding” via data perturbation, for time series
- Perturbation adapts to data properties
  - Automatically combines “random” and “deterministic” at appropriate scales
- Evaluate against both
  - Filtering
  - True value leaks
- Suitable for on-the-fly, streaming perturbation

Overview

- Definitions
- Method
- Experiments
- Conclusion

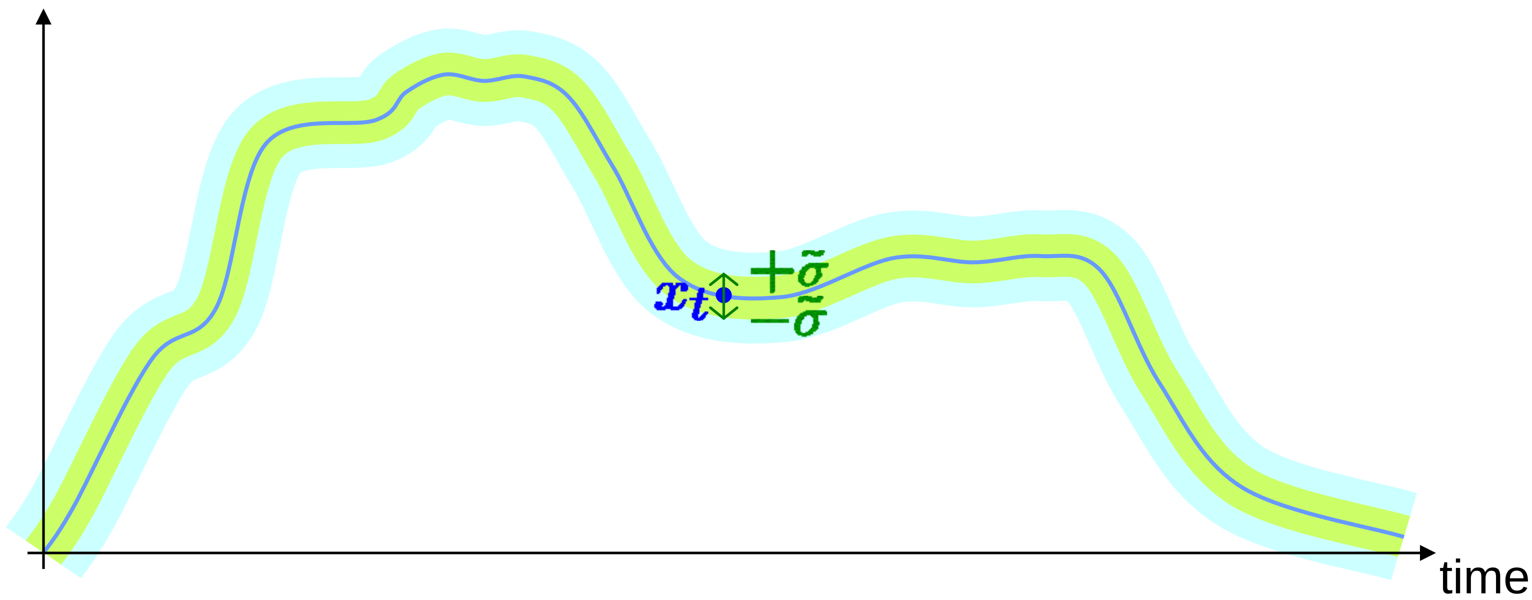
Utility = discord



- Published values y_t are (on expectation) within $\pm\sigma$ of the true values x_t :

$$\mathrm{Var}[y_t - x_t] = \sigma^2$$

Privacy = final uncertainty



- Recovered values $\tilde{x}_t = f(y_t)$ are (on expectation) within $\pm\tilde{\sigma}$ of the true values x_t:

$$\mathrm{Var}[\tilde{x}_t - x_t] = \tilde{\sigma}^2$$

Goal

- Recovery of true values is based on assumptions about attack model, with specific background knowledge
  - Linear filtering
  - Linear reconstruction (based on true values)

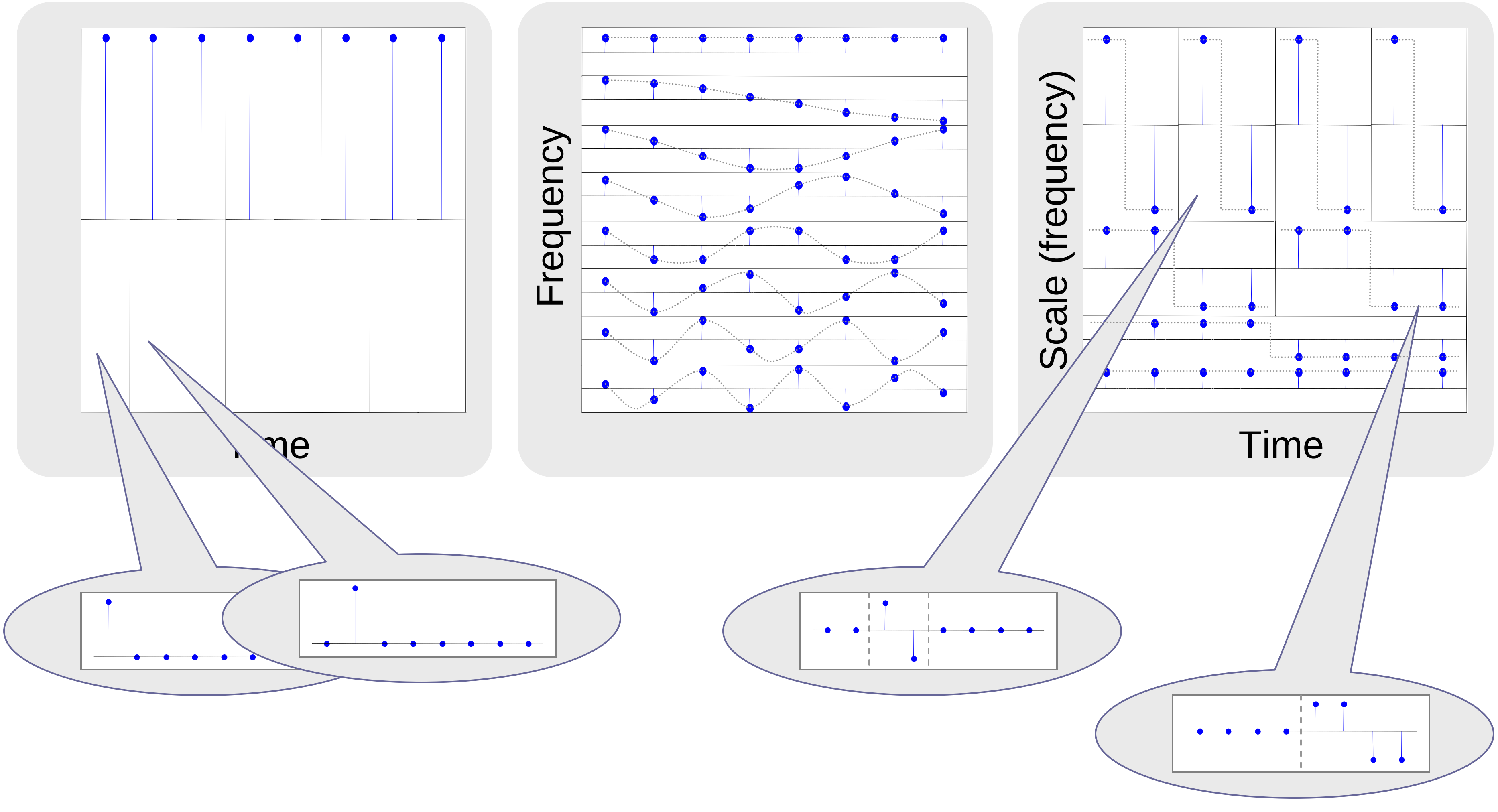
- Goal: $\tilde{\sigma} = \sigma$

Overview

- Definitions
- Method
- Experiments
- Conclusion

Wavelet and Fourier representations

One-slide refresher

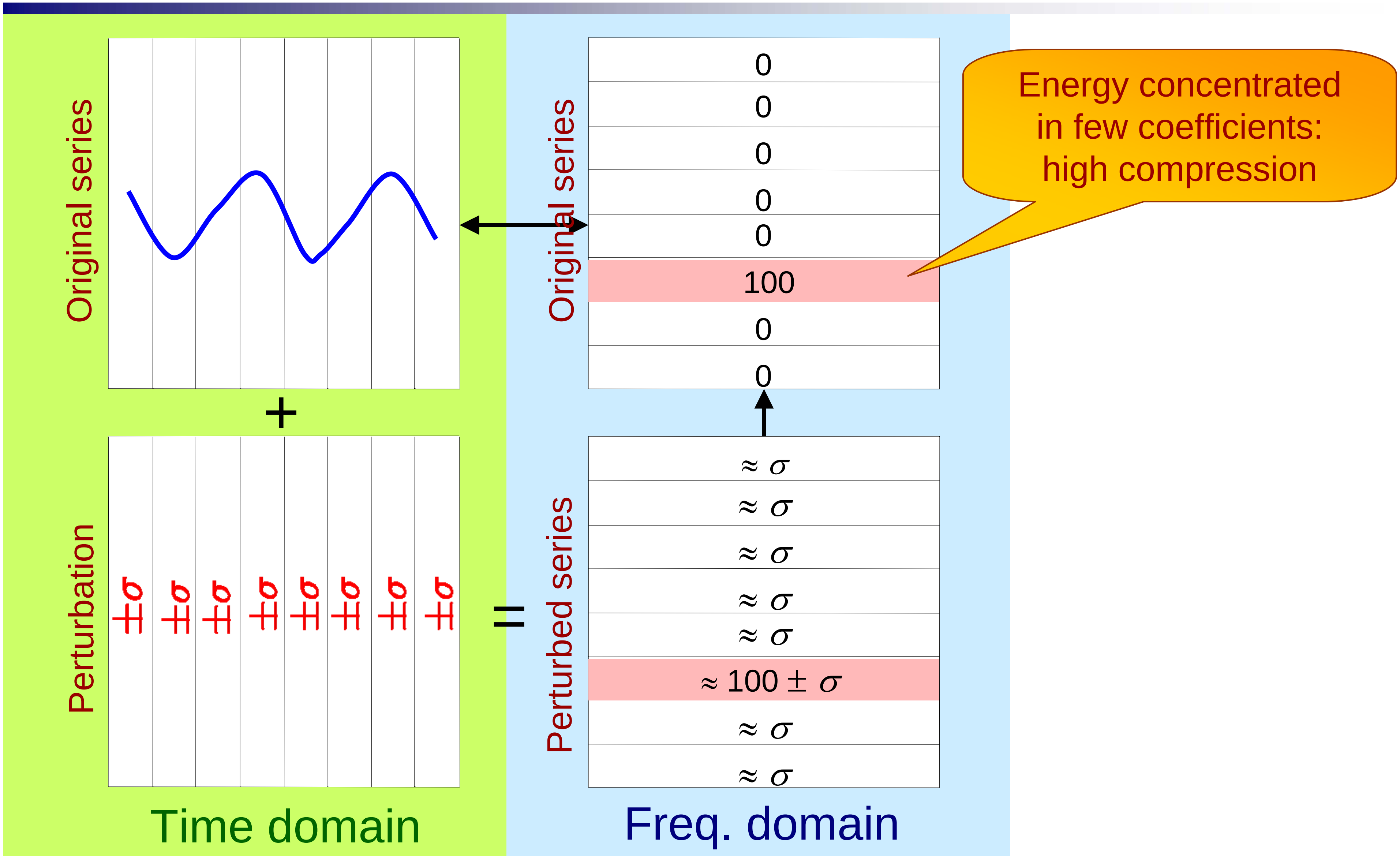


Our work

- Fourier-based perturbation
  - Batch
- Wavelet-based perturbation
  - Batch
  - Streaming

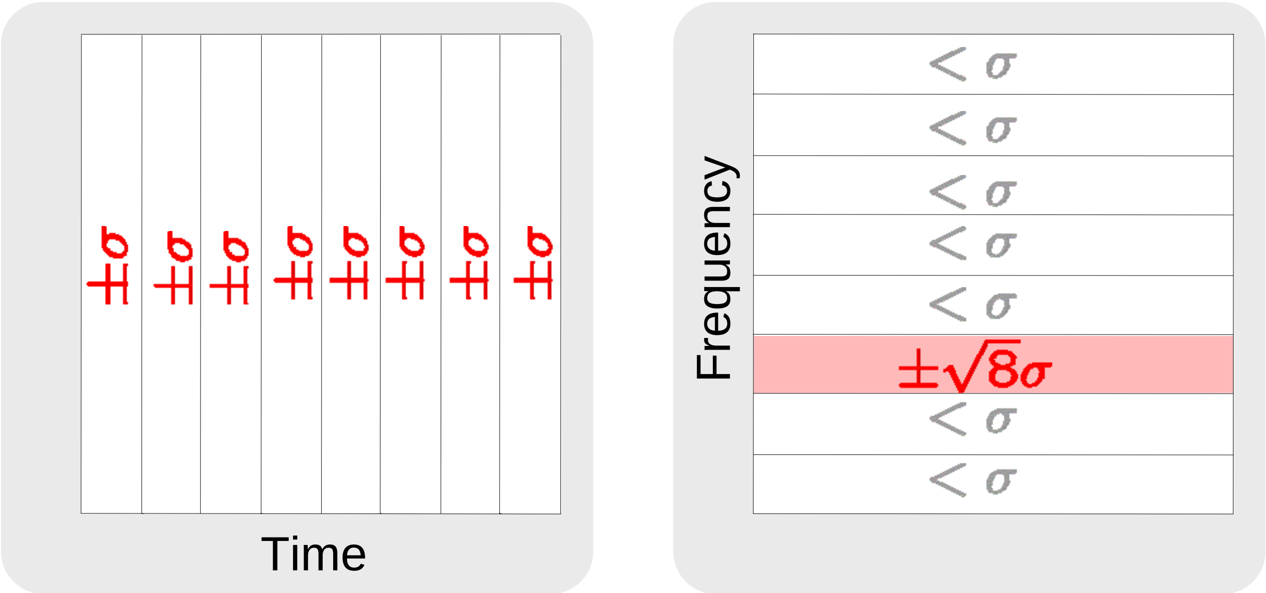
Fourier-based perturbation

Intuition



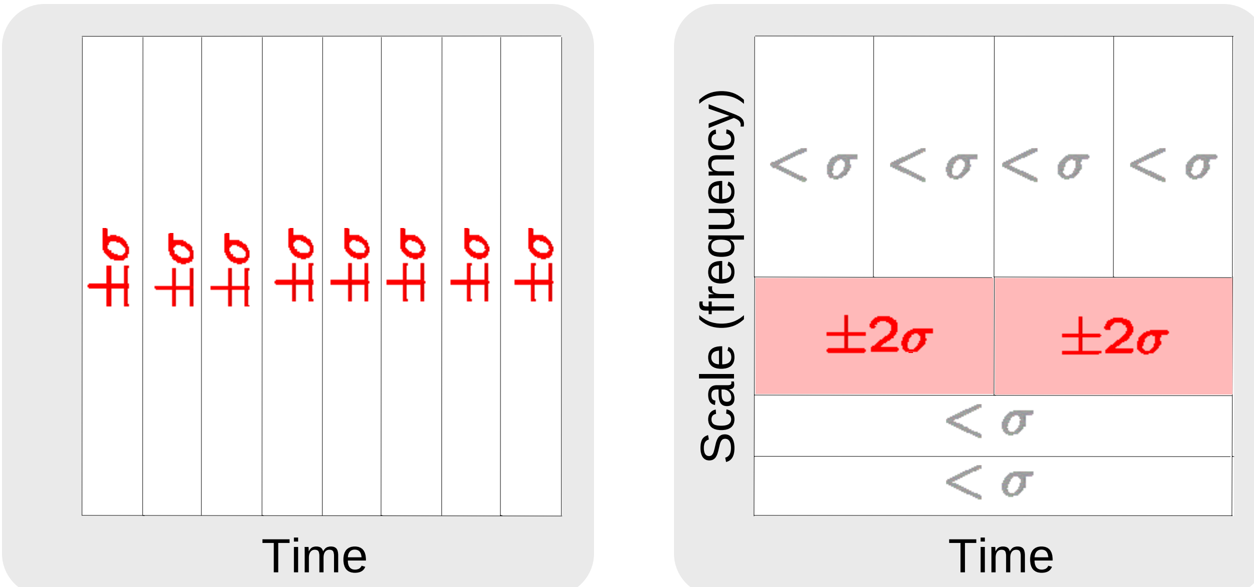
Fourier-based perturbation

Intuition & Summary



Wavelet-based perturbation

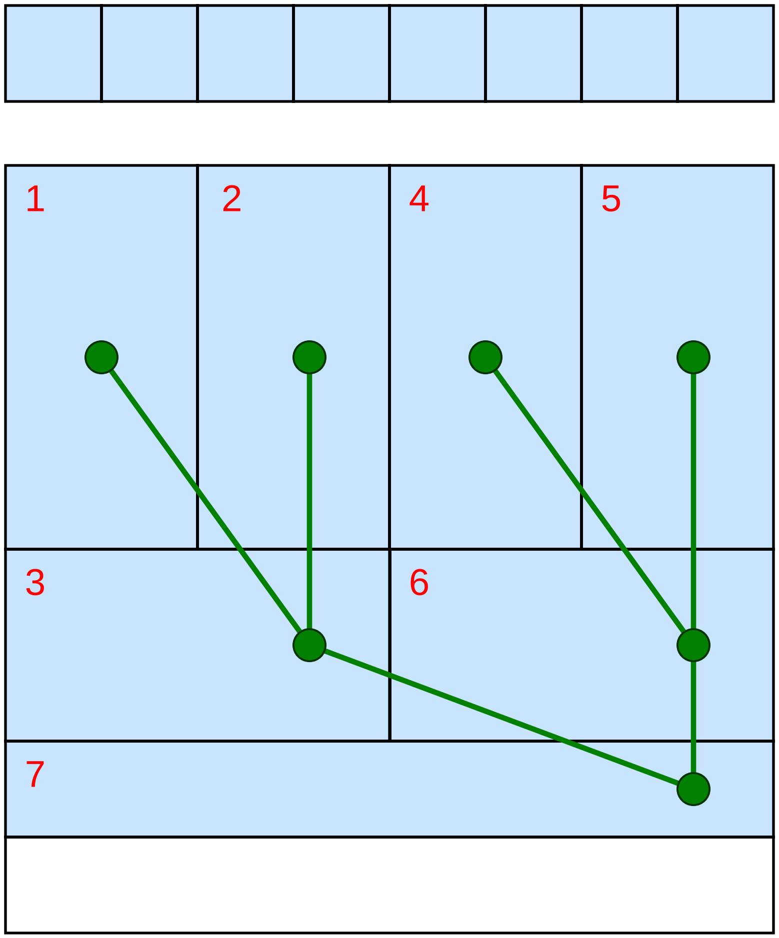
Intuition & Summary



Next: How to do this online?
(1) Wavelet transform; (2) Noise allocation

Streaming perturbation

(1) Wavelet transform—Summary



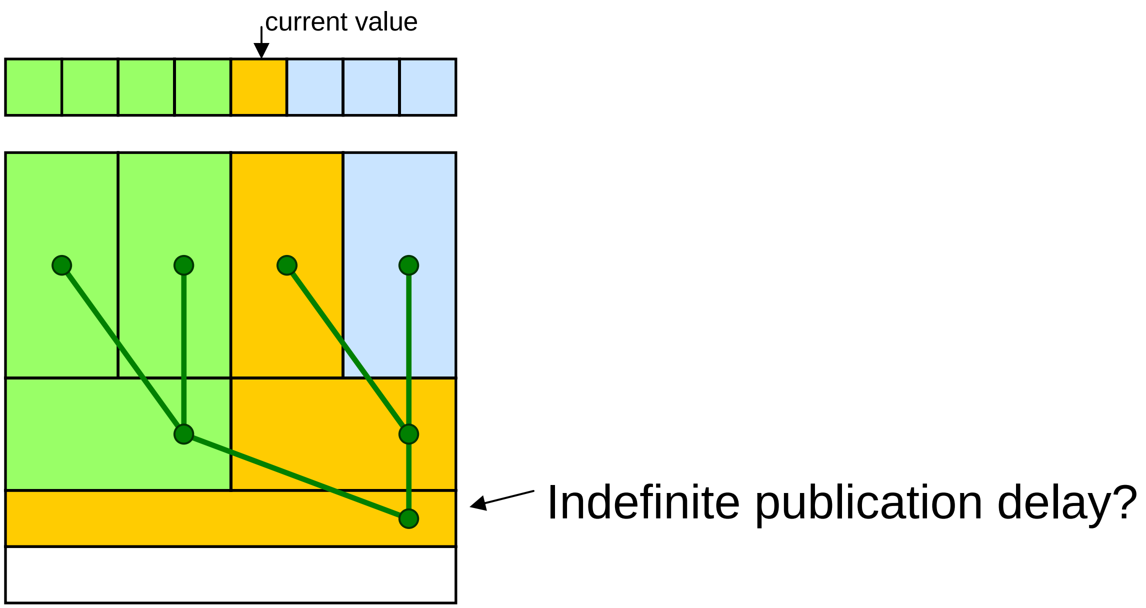
- Forward transform: post-order traversal
- O(lgN) space
- O(1) time (amortized)

Streaming perturbation

(2) Noise allocation—Summary

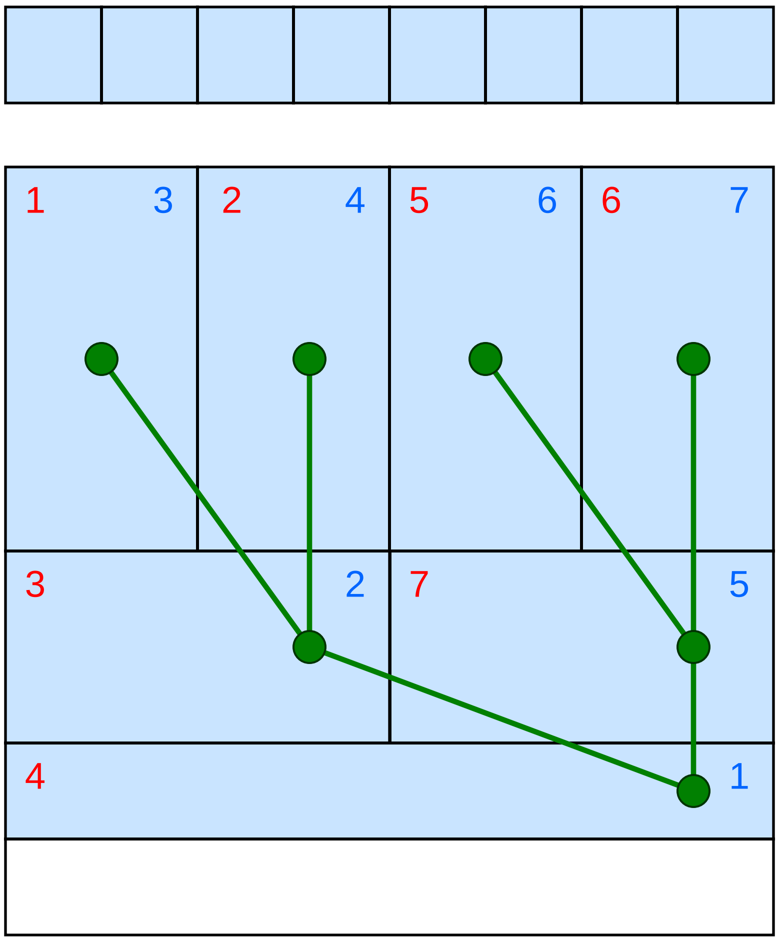
Challenge:

- Knowing **only** the wavelet coefficients up to the current time
- How can we allocate the noise **online** so that it is as close as possible to the batch allocation?



Streaming perturbation

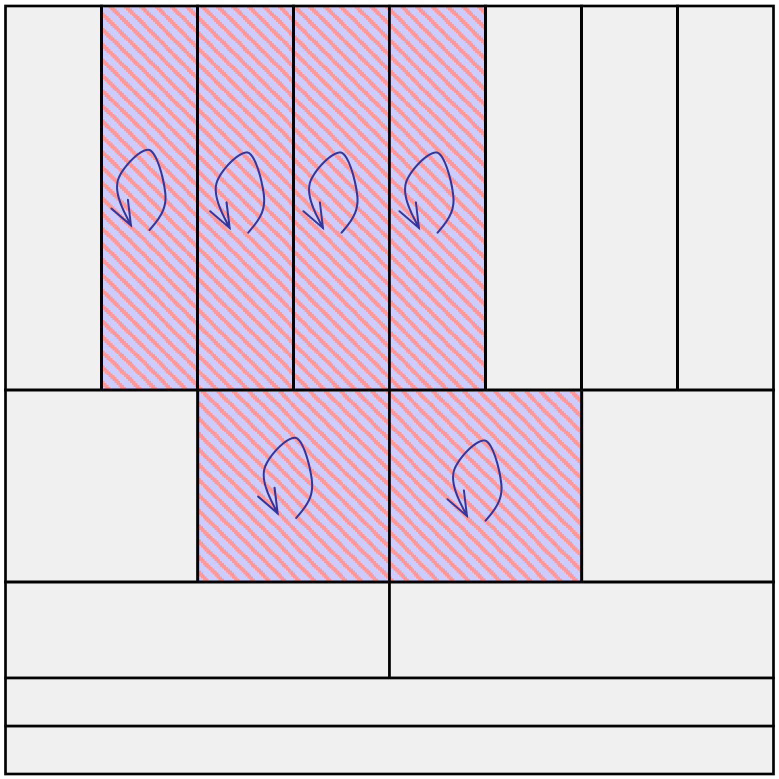
(1) Wavelet transform—Summary



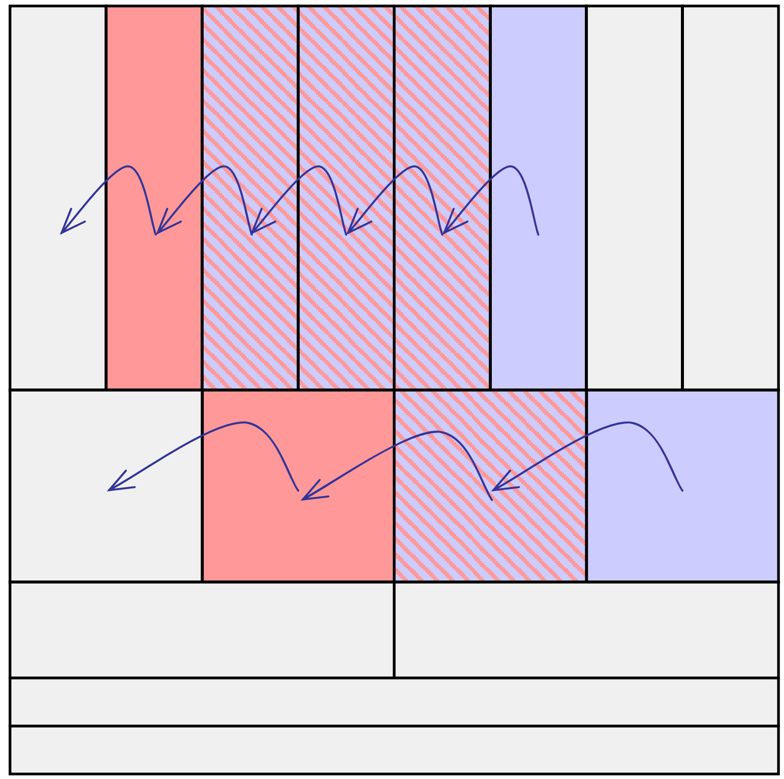
- Inverse transform: pre-order traversal
- O(lgN) space
- O(1) time (amortized)

Streaming perturbation

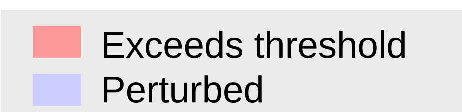
(2) Noise allocation—Summary



Batch



Per-band lookahead



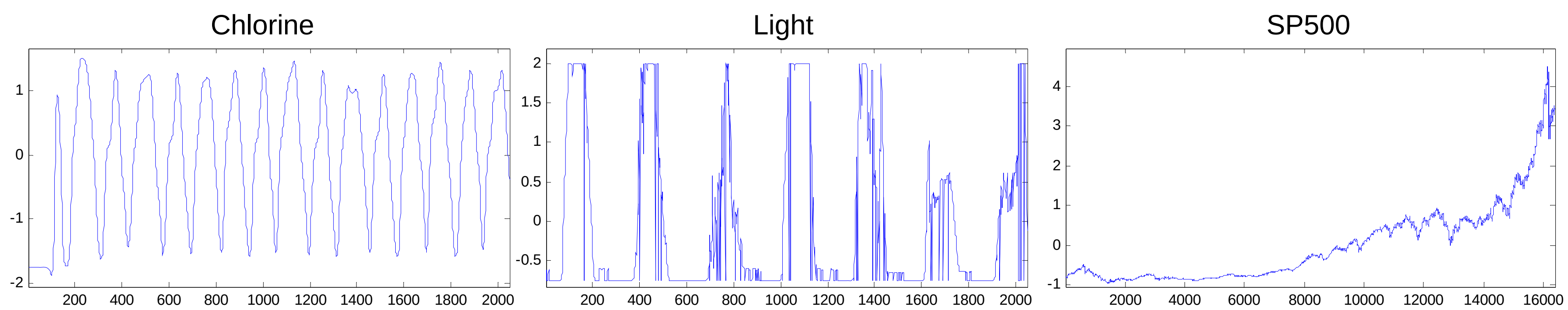
[see paper for details]

Overview

- Definitions
- Method
- Experiments
- Conclusion

Experimental overview

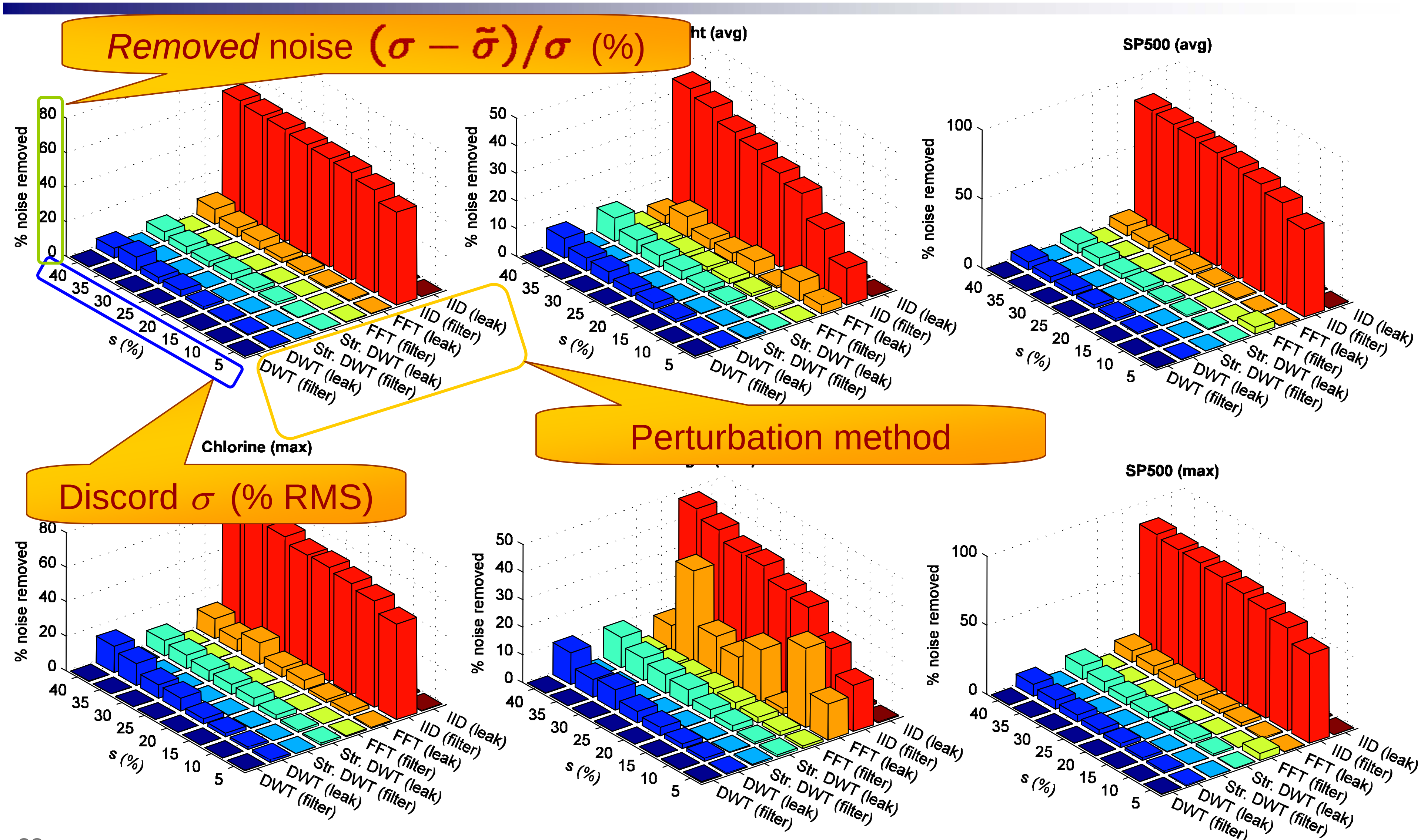
- Datasets:
  - Chlorine: Chlorine concentration in drinkable water distribution network
  - Light: Light intensity measurements (Intel Berkeley)
  - SP500: Standards & Poors 500 index



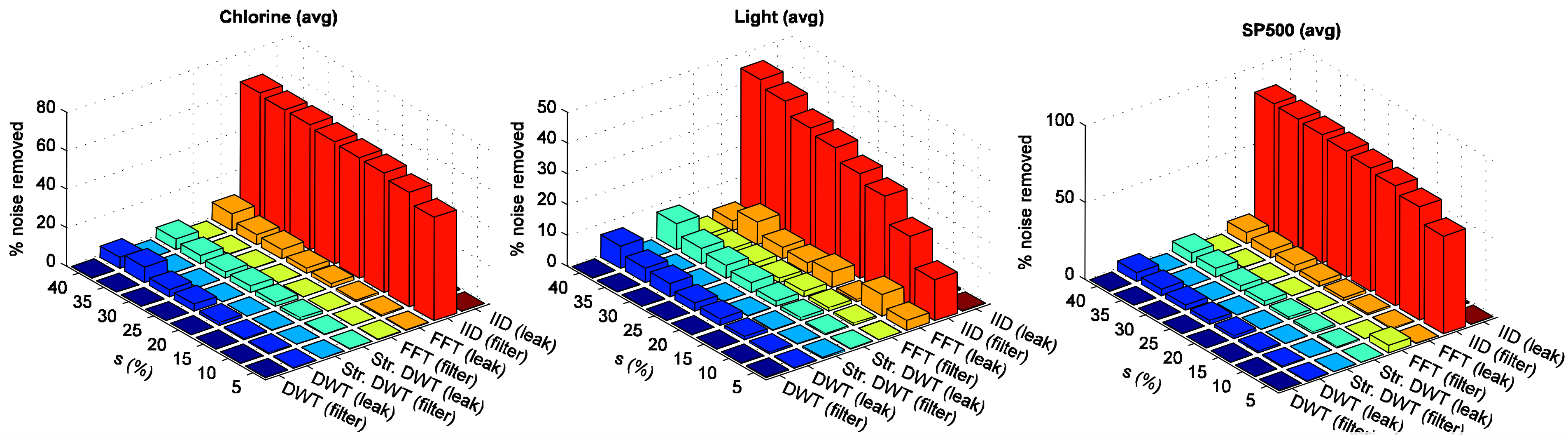
Experimental overview

- Varying
  - Discord levels, and
  - Perturbation methods:
    - IID
    - Fourier-based (FFT)
    - Batch wavelet-based (DWT)
    - Streaming wavelet-based (str. DWT)
- Filter: wavelet shrinkage [Donoho / TOIT95]
- True values: linear regression

Removed uncertainty

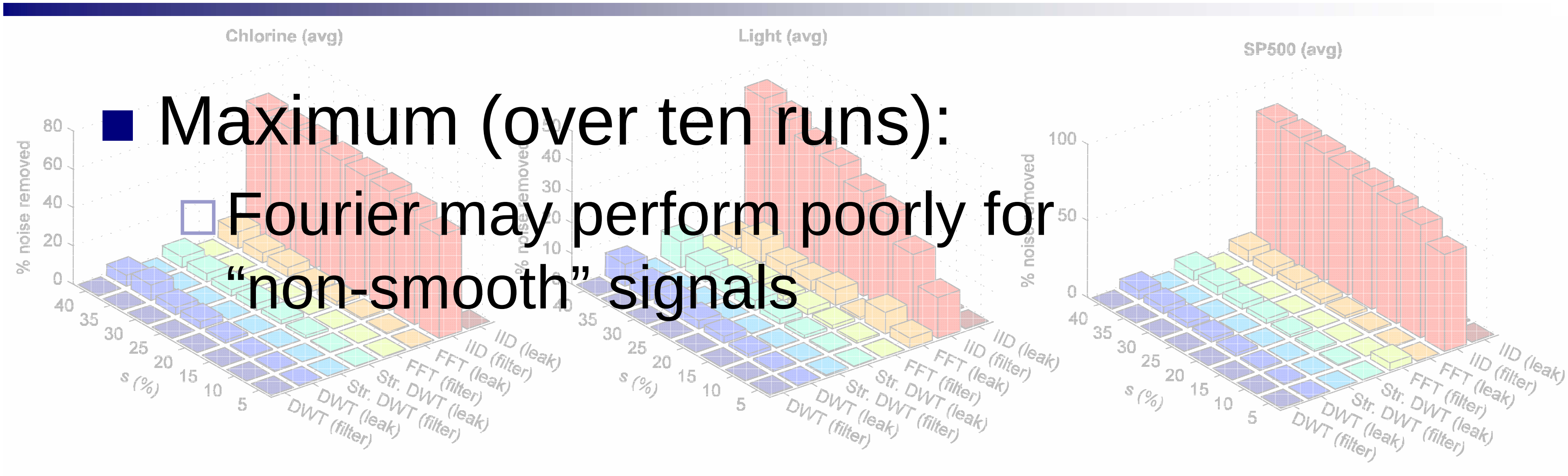


Removed uncertainty

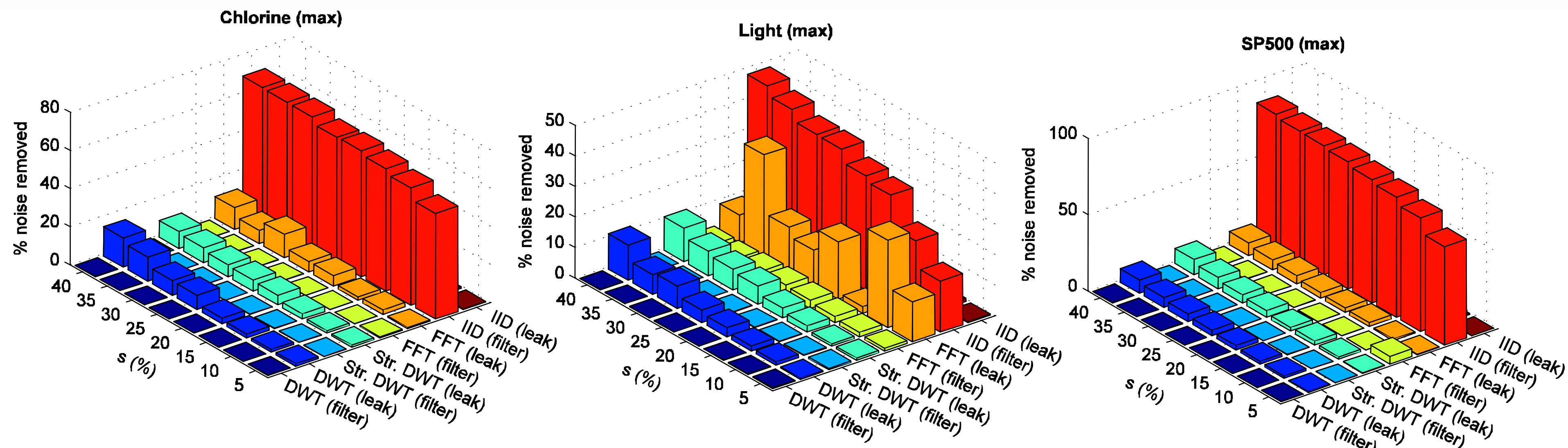


- Average (over ten runs):
  - IID noise: excellent resilience to leaks, very poor for filtering
  - Other methods: comparable

Removed uncertainty

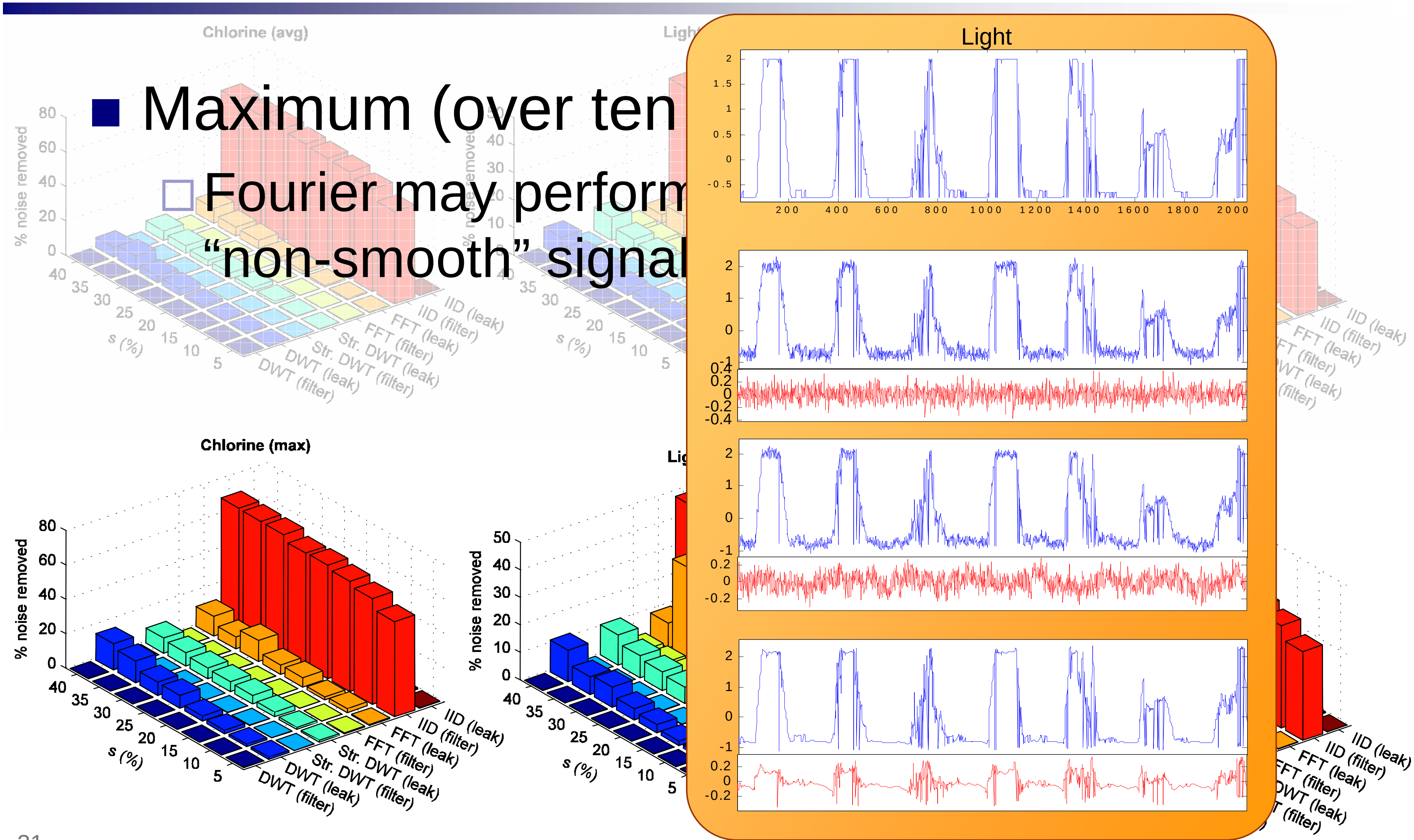


- Maximum (over ten runs):
  - Fourier may perform poorly for "non-smooth" signals

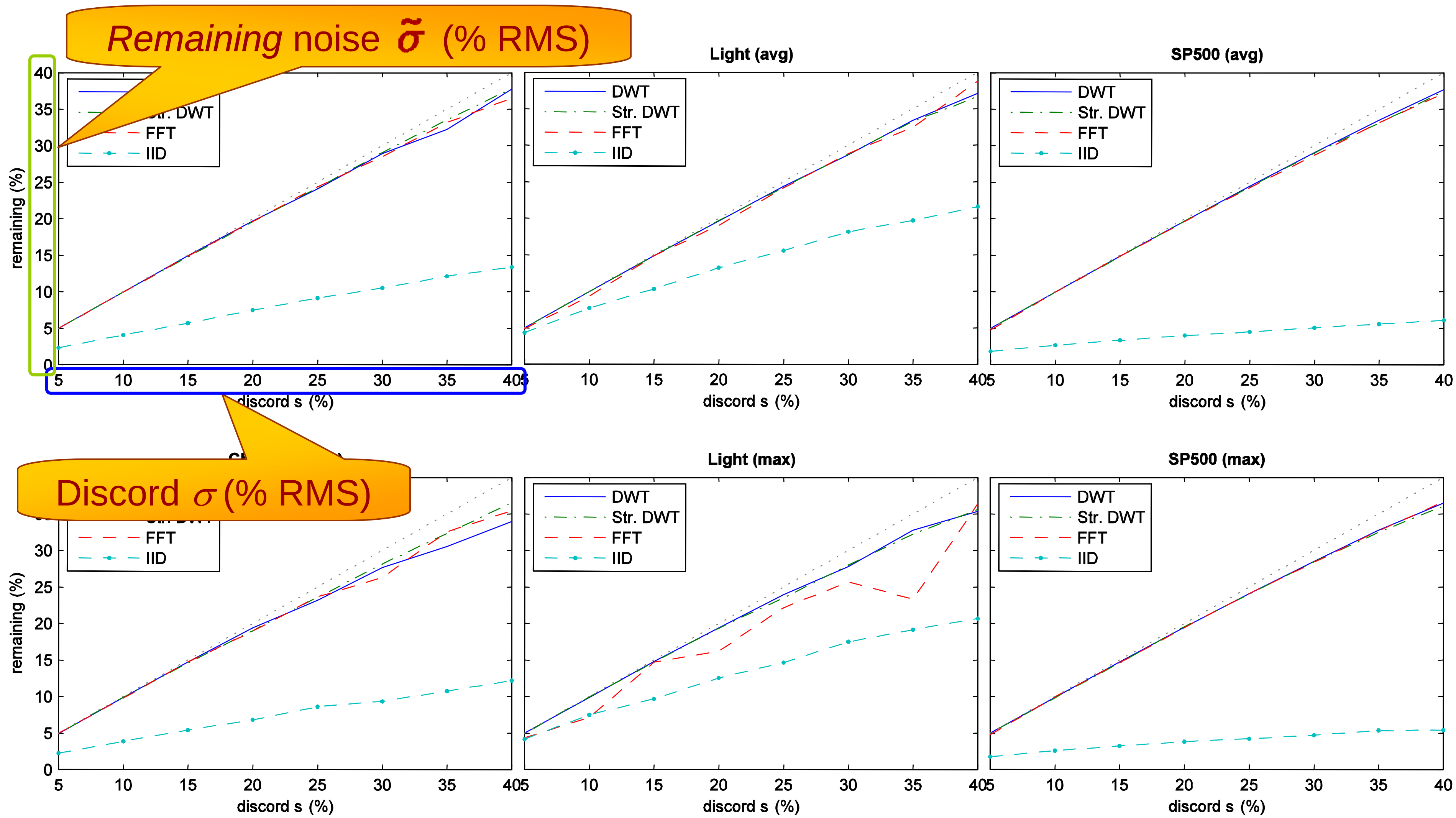


Removed uncertainty

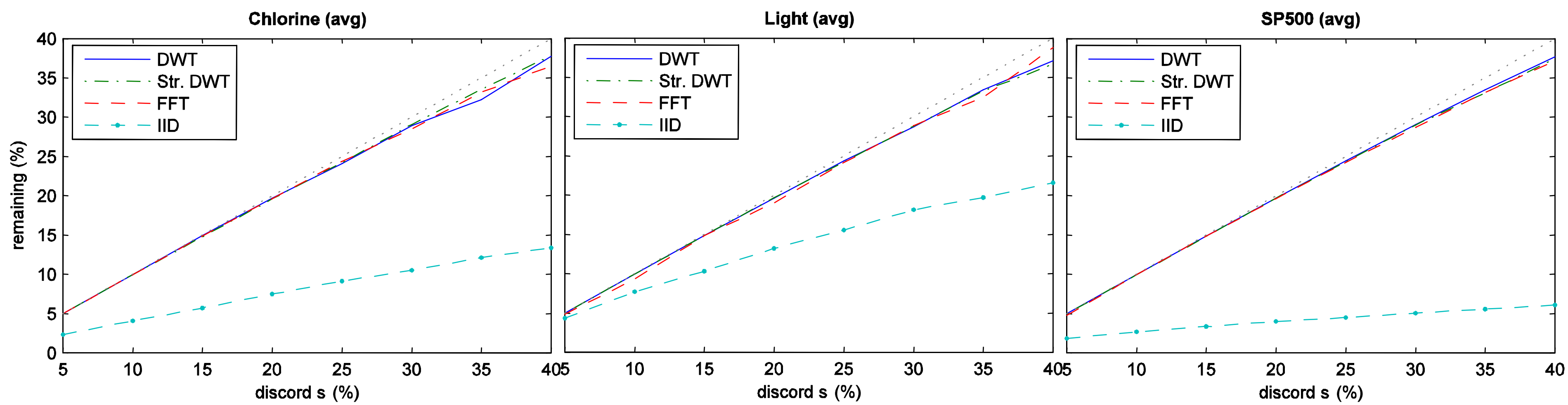
- Maximum (over ten
  - Fourier may perform
    "non-smooth" signal



“True” uncertainty



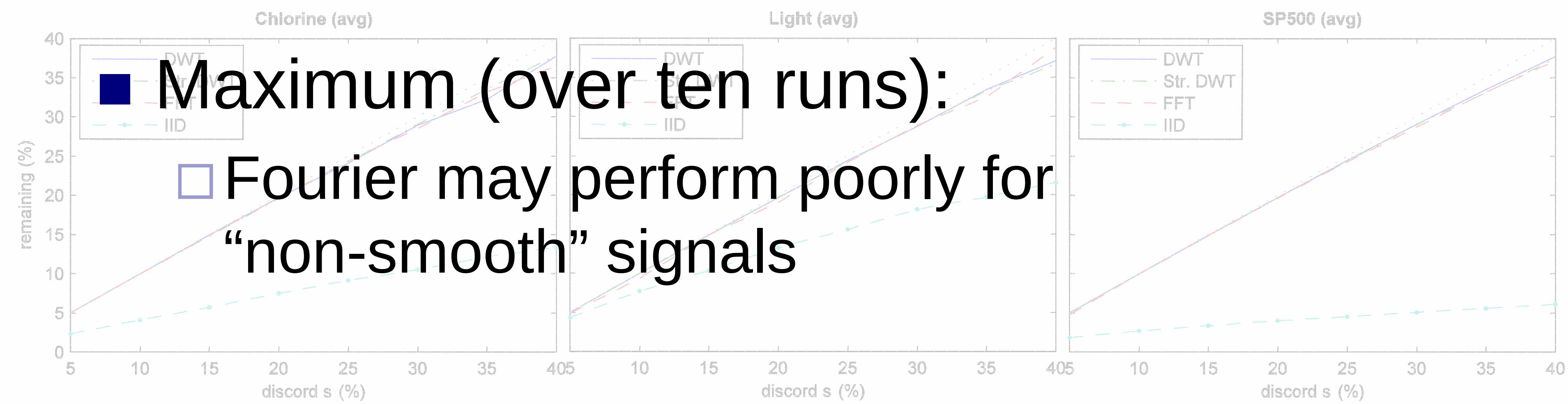
“True” uncertainty

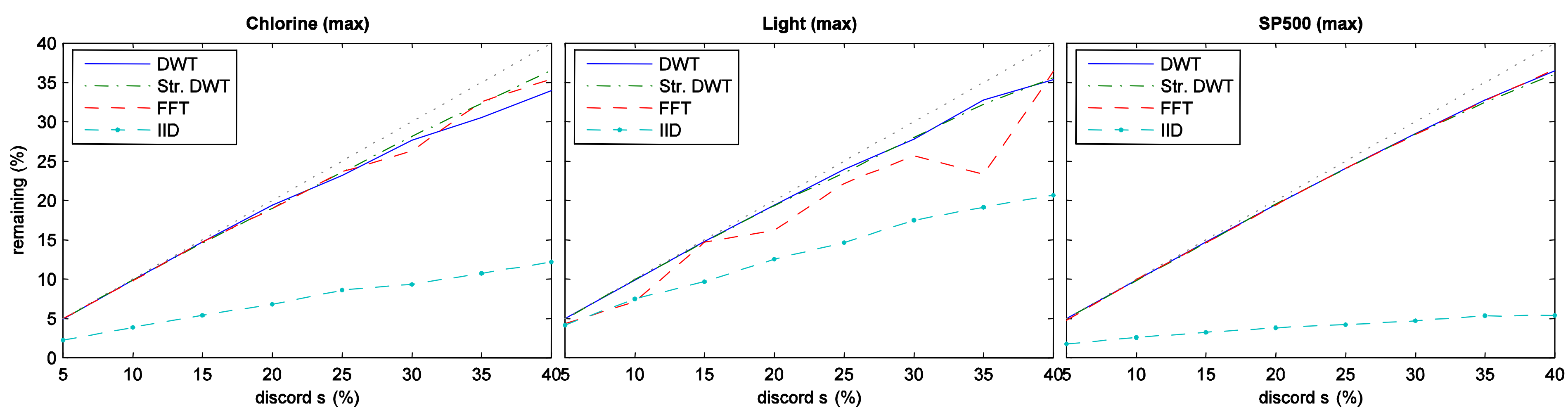


- Average (over ten runs):
  - IID noise: very poor overall
  - Other methods: comparable

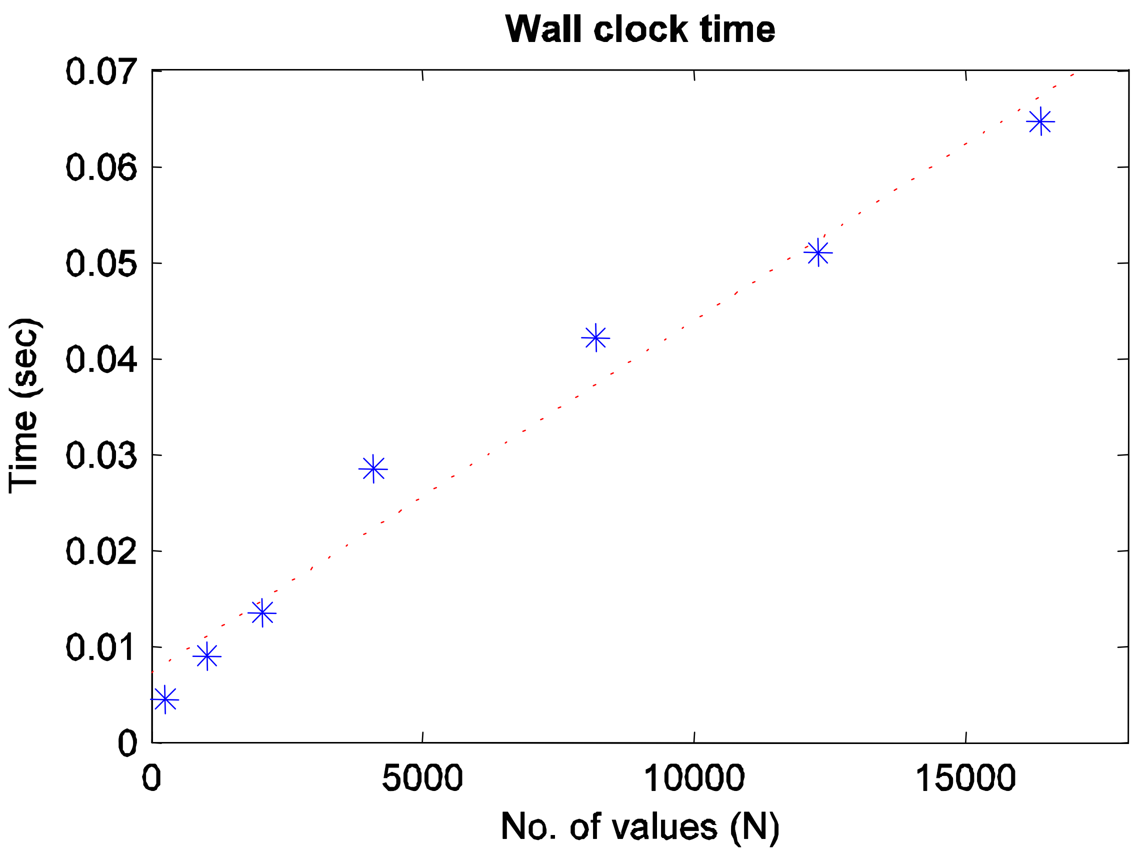
“True” uncertainty

- Maximum (over ten runs):
  - Fourier may perform poorly for “non-smooth” signals





Scalability



Constant per measurement

Overview

- Definitions
- Method
- Experiments
- Conclusion

Related work (1/2)

- **Privacy-preserving data mining**
  - **SMC** [Lindel & Pinkas / CRYPTO00], [Vaidya & Clifton / KDD02]
  - **Partial information hiding**
    - **Perturbation** [Agrawal & Srikant / SIGMOD00], [Du & Zhan / KDD03], [Kargupta, Datta, Wang & Sivakumar / ICDM03], [Agrawal & Aggarwal / EDBT04], [Chen & Liu / ICDM05], [Huang, Du & Chen / SIGMOD05], [Liu, Ryan & Kargupta / TKDE05], [Li et al. / ICDE07]
    - ***k*-anonymity** [Sweeney / IJUFKS02] , [Aggarwal & Yu / EDBT04], [Bertino, Ooi, Yang & Deng / ICDE05], [Kifer & Gehrke / SIGMOD06], [Machanwajjala, Gehrke & Kifer / ICDE06], [Xiao & Tao / SIGMOD06]
  - **Interactive privacy** [Blum, Dwork, McSherry & Nissim / PODS05], [Dwork, McSherry, Nissim, Smith / TCC06]
    - SSDBs [Denning / TODS80]
- **Wavelets in DM** [Gilbert, Kotidis, Muthukrishnan & Strauss / VLDB01], [Garofalakis & Gibbons / SIGMOD02], [Bulut & Singh / ICDE03], [Papadimitriou, Brockwell & Faloutsos / VLDB04], [Lin, Vlachos, Keogh & Gunopulos / EDBT04], [Karras & Mamoulis / VLDB05]
- **Compression and DM** [Keogh, Lonardi & Ratanamahatana / KDD04]

Related work (2/2)

- Correlated perturbation [Kargupta, Datta, Wang & Sivakumar / ICDE03], [Huang, Du & Chen / SIGMOD05], for streams [Li et al. / ICDE07]

- L-diversity [Machanwajjala, Gehrke & Kifer / ICDE06] and personalized privacy [Xiao & Tao / SIGMOD06]
- Dimensionality curse and privacy [Aggarwal / VLDB05]

- Watermarking [Sion, Attalah & Prabhakar / TKDE06]
- Compressed sensing [Donoho / TOIT06], [Candés, Romberg & Tao / TOIT06]

Conclusion

- Partial information hiding via data perturbation
- User-defined discord (utility)
- Adapts to data properties
  - Automatically combines “random” and “deterministic” at appropriate scales
  - Additionally preserves spectral properties
- Evaluate against both
  - Filtering
  - True value leaks
- Suitable for on-the-fly, streaming perturbation

Perturbing data objects with any “structure” is non-trivial, even under fixed attack model(s)

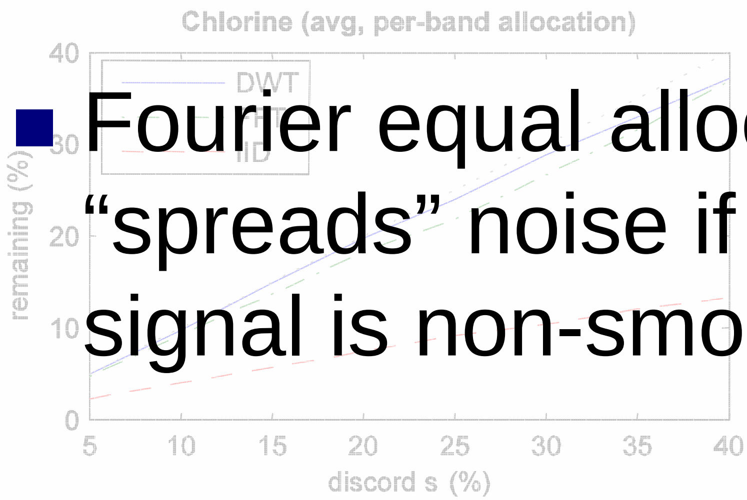
Thank you

Time Series Compressibility and Privacy

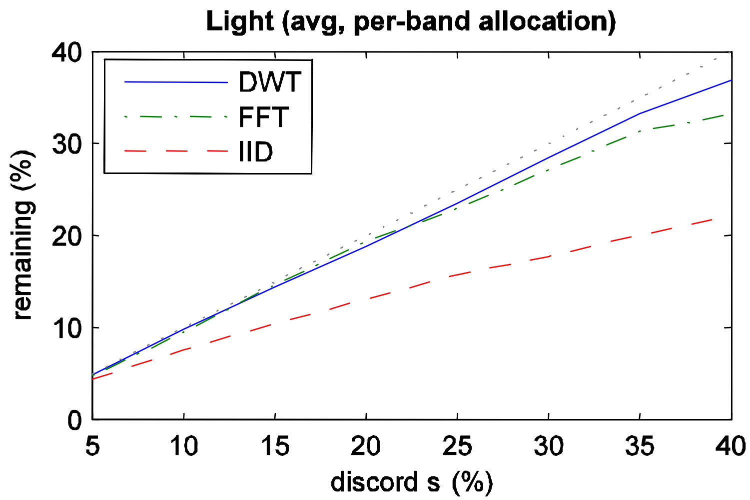
Spiros Papadimitriou*
Feifei Li+
George Kollios+
Philip S. Yu*

**IBM TJ Watson*
+Boston University

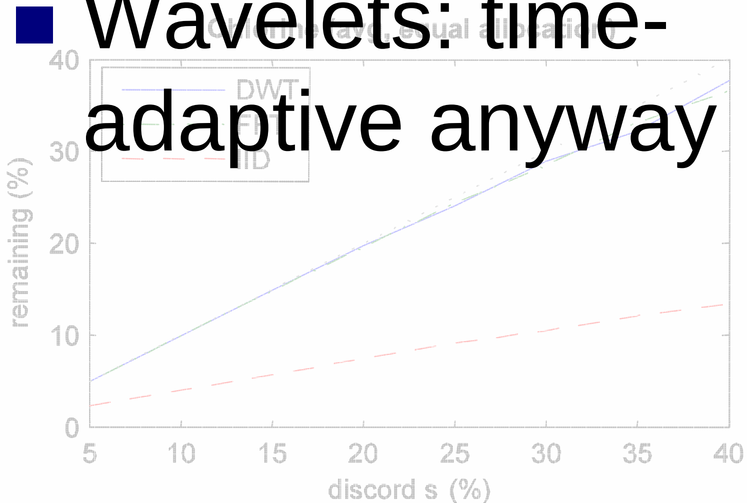
Per-band allocation

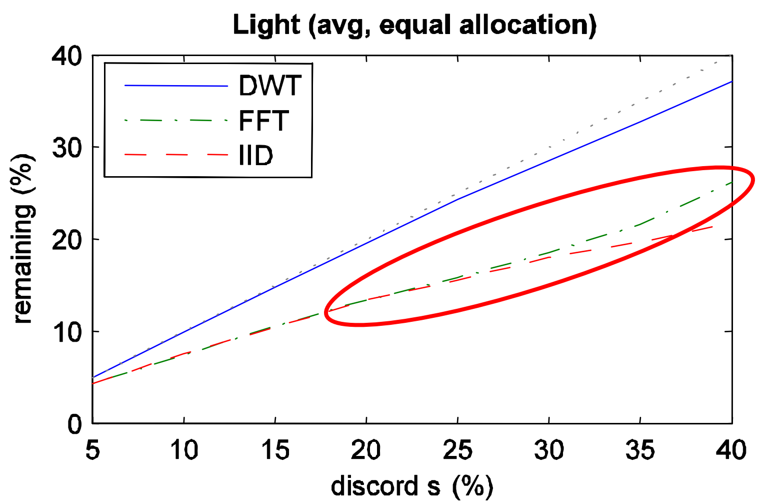


- Fourier equal alloc.: "spreads" noise if signal is non-smooth

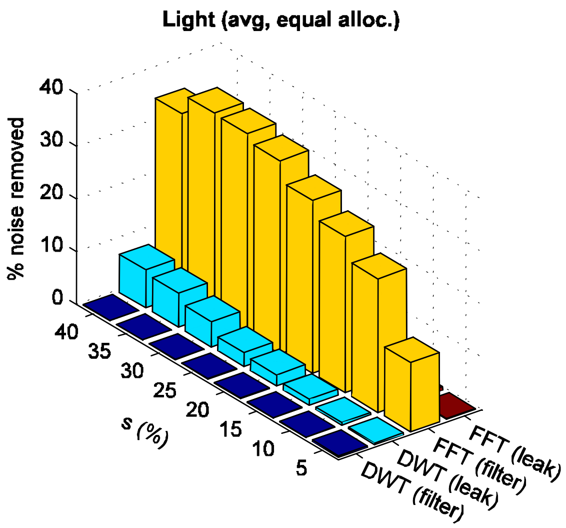


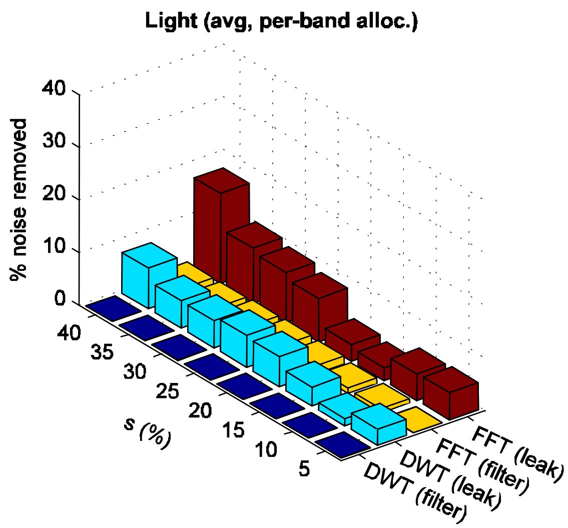
- Wavelets: time-adaptive anyway



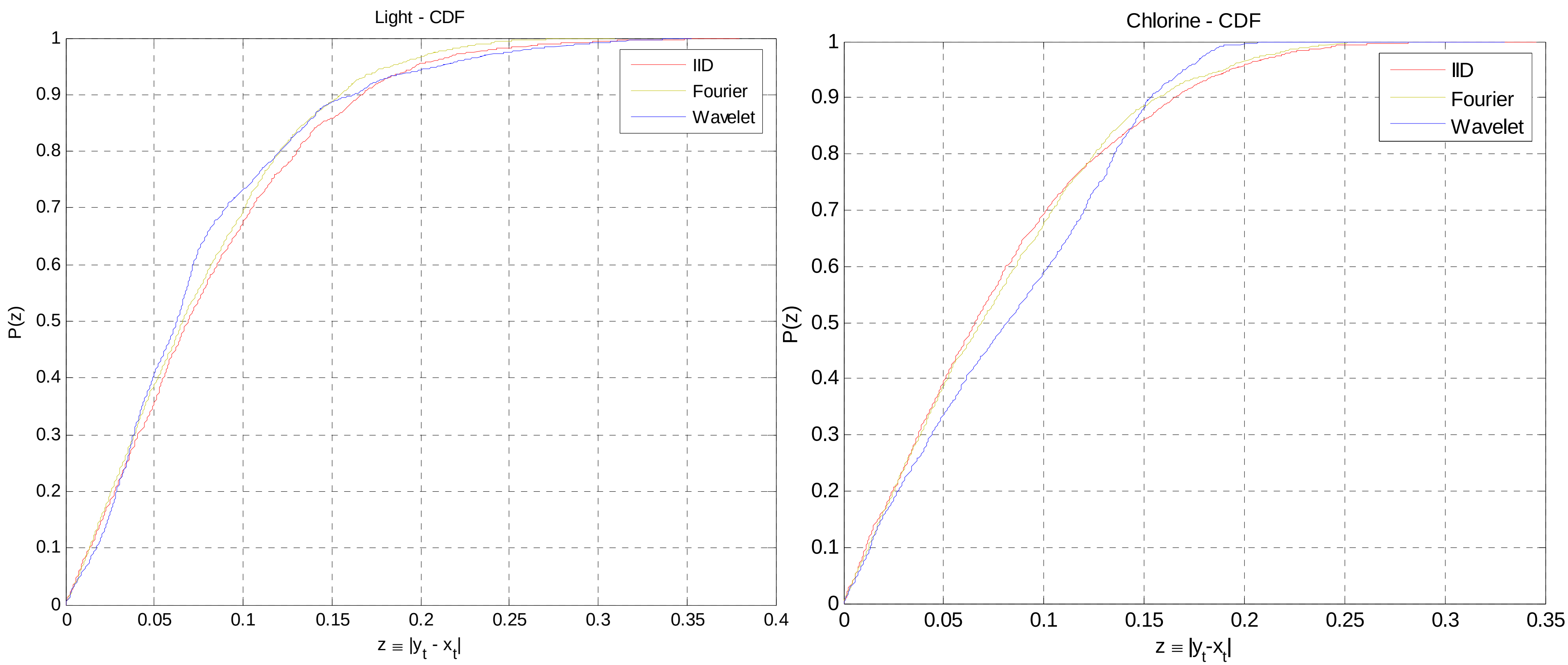


Per-band allocation





Marginals



Light - CDF
IID
Fourier
Wavelet
P(z)
$z \equiv |y_t - x_t|$
Chlorine - CDF
IID
Fourier
Wavelet
P(z)
$z \equiv |y_t-x_t|$